LETTER

Real-Time Touch Controller with High-Speed Touch Accelerator for Large-Sized Touch Screens

SangHyuck BAE[†,††], ***Member***, **DoYoung JUNG**[††], **CheolSe KIM**[††], **KyoungMoon LIM**[††], ***Nonmembers***, ***and*** **Yong-Surk LEE**[†a)], ***Member***

SUMMARY For a large-sized touch screen, we designed and evaluated a real-time touch microarchitecture using a field-programmable gate array (FPGA). A high-speed hardware accelerator based on a parallel touch algorithm is suggested and implemented in this letter. The touch controller also has a timing control unit and an analog digital convert (ADC) control unit for analog touch sensing circuits. Measurement results of processing time showed that the touch controller with its proposed microarchitecture is five times faster than the 32-bit reduced instruction set computer (RISC) processor without the touch accelerator.

key words: *real-time, FPGA, parallel processing, digital filter, touch controller, capacitive touch*

1. Introduction

Since the projected capacitive touch screen utilizes mutual capacitance that changes with touch, the screen has advantages such as thinness, transparency, softness, and use in personal electronic devices [1]. As the size of capacitive touch screen increases, touch screens are required to maintain the same high performance that is expected from a smartphone but with a lager screen. This means that the touch screen must scan more touch sensors in the same amount of time. In addition, the processor in the touch screen has to process with more touch workloads while still maintaining an acceptable processing time, report rate, and power efficiency. Current capacitive touch screens used in smart mobile devices that have 200–400 capacitive nodes typically take a 60–100 Hz report rate and 10–16.7 ms to update the touch positions in response to a physical touch. For a positive user experience, it is important to maintain real-time reporting of the touch positions in a touch application. This means that the processor should guarantee a touch response within 10 ms. In a touch application, the touch module has a separated processor from the main high-performance processor or application processor in a smartphone. However, the power consumption of a touch module is limited to a few tens of mW, and a slowing clock rate is required to reduce power consumption. The processing unit in the touch screen application has different tasks such as control tasks and data processing tasks. Control tasks are timing control for touch sensors, readout circuits, and system interfacing with touch position results. Data processing tasks include data filtering for the raw touch data and touch-position calculations. In a touch screen module that is larger than 10 inches, processors become more complex owing to the increased number of arithmetic tasks such as matrix operations and floating point arithmetic that are used in touch-position calculations and noise filtering. This has increased the demand for higher computation power and low-power consumption in the mobile environment.

We propose a hardware touch accelerator architecture, with low power consumption, to reduce the processing time of large-sized touch screens.

2. Microarchitecture for the Touch Controller

The architecture of the proposed touch controller with hardware accelerator is shown in Fig. 1. The system consists of a scalar processor, a touch accelerator, a high-speed block-

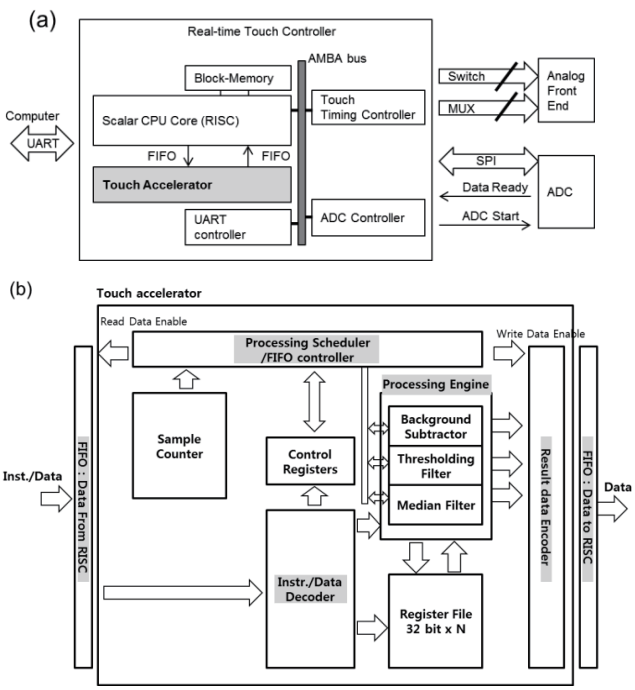

Fig. 1 Structure of suggested (a) real-time touch controller and (b) touch accelerator.

Manuscript received July 22, 2014.
Manuscript revised September 18, 2014.
Manuscript publicized October 17, 2014.

[†]The authors are with School of Electrical & Electronic Engineering, Yonsei University, Seoul, 120–749 Korea.
[††]The authors are with R&D Center, LG Display Co. Ltd., Paju-city, 413–811 Korea.
a) E-mail: yonglee@yonsei.ac.kr

DOI: 10.1587/transinf.2014EDL8152

memory, and peripheral units. The RISC core performs scalar operations and peripheral control. The touch accelerator oversees image-processing filtering operations. The system is implemented on a Xilinx FPGA. The touch controller is designed to have a one clock latency memory system using block random access memory (RAM) for minimum delay. Block memories are used for storing data in the scalar core. To reduce processing time, multiple steps of algorithm are processed in the processor. The accelerator processes matrix data without repeatable load and store operations. To transfer the data to the accelerator, first-in-first-out (FIFO) buffers are used. The maximum transfer rate of FIFO is 300 million words per second. The data width and the architecture of the accelerating processor are optimized for a touch panel that was presented in our previous work [2]. The accelerator has processing element units and they have registers, a counter, and a multiplexer controller.

2.1 Scalar Processor

The system contains a RISC 32-bit soft processor core that is based on the Microblaze architecture. The MicroBlaze is based on the Harvard architecture, and support the Fast Simplex Link (FSL) for the interface with the hardware accelerator. Instruction/data can be transferred using dedicated C language instructions. The RISC core controls the operations of the driving/sensing blocks (touch timing controller) and to acquire the raw data from touch screen (ADC controller). The RISC core also performs scalar operations such as setting threshold level, touch position calculating, and host interfacing.

2.2 Touch Accelerator

The touch accelerator has multiple processing units for touch matrix data input. For high-speed processing, the accelerator uses multiple functional units in parallel. The simultaneous execution of independent instructions such as background subtraction, thresholding, and calculating the median are shown in Fig. 2. As reported in previous work [2], the touch algorithm is composed of preprocessing of the raw touch data, calculating touch positions using matrix results, and post processing for the user experience interface such as a smoothing algorithm. The preprocessing algorithm uses background subtraction (B/S) for touch detection and data scaling, median filtering for noise rejection, and thresholding for labeling algorithm. Median filtering is very effective to reduce noise in digital image processing, and it also shows good performance in touch applications. When the median filter is realized by software algorithm in the RISC processor, much processing time is spent. For high-speed processing of median filtering, a hardware implementation method that has minimal latency is suggested by Lee et al. [3] and Chen et al. [4]. In the median filter architecture, input data is compared and sorted using a parallel method at each processing element (PE) unit. The configuration table decides whether each new sample will be

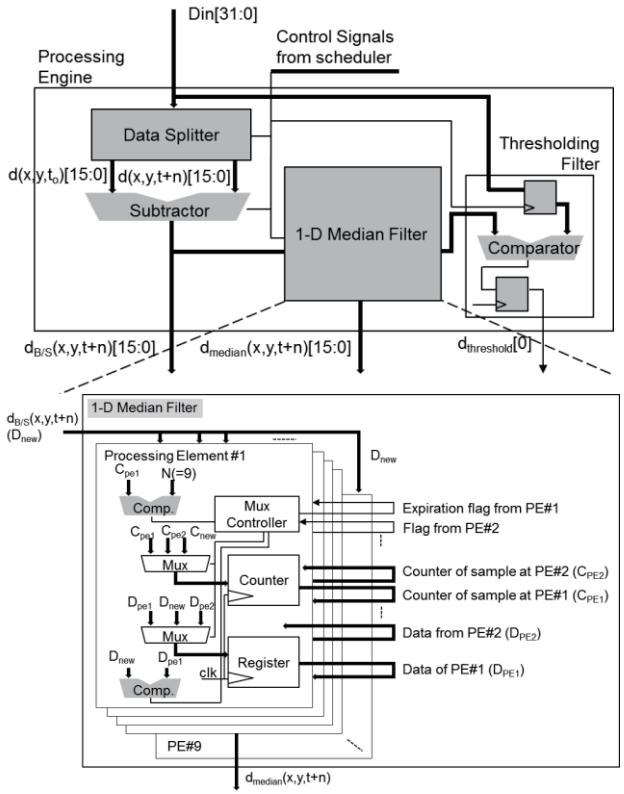

Fig. 2 Architecture of the touch processing engine.

inserted into the register in PE or not within same clock of data input. We extended the idea in [4] by proposing a parallel-processing architecture for touch application.

The touch accelerator has nine processing element units for median filtering of a 3×3 matrix window. Each processing element has registers, comparators, multiplexer configuration tables, and buses for data/counter transfers to other processing units. For parallel processing the raw touch data, the scalar processor sends prerecorded touch frame data and current touch data simultaneously. Data decoder in the accelerator splits 32-bit data into 16-bit background and 16-bit touch data. For the threshold level, the data decoder makes a register control signal for a threshold level register when threshold value is transferred from the scalar processor. The data encoder merges 16-bit median output (or B/S result) and threshold data. The resultant data at the touch accelerator is ready just at the next clock after putting the raw data. For the median filtering throughput, three clocks are required for each nine (3×3 median window) inputs, and nine clocks are required for the first window of each line. All of the sequence is controlled by the scheduler. Performance improves by calculating the median, B/S, and the thresholding algorithm of touch data using a proposed parallel method, as shown in Fig. 2.

The median processing sequence in the touch accelerator is shown in Fig. 3. Touch data is transferred sequentially from the RISC processor, the $(I + 1)^{th}$ window reuses from the I^{th} window data, and only three clocks are required for

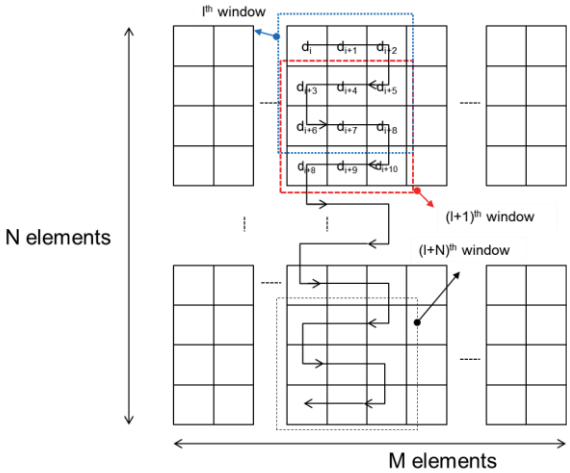

Fig. 3 Data flow and processing sequence in the touch accelerator.

each nine median window data. For the $(I+1)^{th}$ window processing, d_i, d_{i+1}, and d_{i+2} are removed sequentially because their counter value are over nine.

The first step of the accelerator, background subtraction algorithm is calculated as shown in Eq. (1).

$$d(t+n) = d(x, y, t+n) - d(x, y, t_o) \tag{1}$$

Current sensed data d(x, y, t + n) is subtracted from the pre-recorded touch data d(x, y, t0). B/S data $d_{B/S}$ is sent to the encoder, median filter. The comparator decides whether $d_{median}(x, y, t + n)$ is bigger than pre-transferred threshold level, the writes d_{thresh} as one or zero. After three data are input into the accelerator for a new window, the encoder latches the median and threshold result, as shown in Eq. (2), and send concatenated data to the RISC processor, as shown in Eq. (3).

$$d_{median} = m[d_{biggest}, d_{2nd}, \ldots, \underline{d_{5th}}, \ldots, d_{smallest}] \tag{2}$$

$$d_{out} = d_{threshold}\ \&\ (d_{B/S}\ or\ d_{median}) \tag{3}$$

The encoder write $d_{B/S}(x, y, t+n)$ instead of $d_{median}(x, y, t+n)$ when B/S and thresholed results are required.

2.3 Panel, Analog Circuits and Peripheral Controller Design

For the touch panel, we designed the equivalent panel load on printed circuit boards. The panel is composed of transmitting sensors, receiving sensors, and a cover plate. Mutual capacitance is formed between these two types of sensors. A finger contacting the screen acts as an electrode and decreases the mutual capacitance by altering the electric charge distributions on the sensor electrodes. When a finger touches the cover plate, Cm is reduced and the sensing circuit converts the variation into voltage. The touch panel has 15×20 touch sensors as shown in Fig. 4.

For the large-size panel evaluation, four successive frames ($15 \times 20 \times 4 = 1{,}200$ touch nodes) are concatenated into one large frame. The analog circuit is composed of excitation and sensing circuits. The excitation circuit generates a square pulse with a high-level voltage. The voltage of the excitation pulse can be changed using a level shifter. The peripheral controller is designed to control analog front end circuits and to interface with an ADC converter. To synchronize the ADC with excitation and sensing circuits, the timing controller generates control signals.

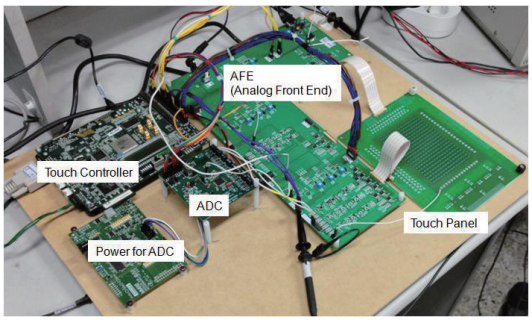

Fig. 4 Prototyping systems for the touch screen controller.

The digitized converted data (capacitance changes of voltage value on the touch panel) from the ADC is transferred using a serial peripheral interface (SPI). All touch processing is carried out in the controller and results are sent to the computer via a UART serial interface.

3. Evaluation Results

The touch controller was implemented on an FPGA for a touch performance evaluation, as shown in Fig. 4.

For the performance evaluation of each filter, results from each step using a frame of 15×20 raw touch data are transferred to computer via UART serial interface, as shown in Fig. 5. Processing time was measured using an internal clock counter. Processing times for a large-size panel that is concatenated four frames are measured five times and average value is listed on Table 1 and total processing steps are shown in Fig. 6. Scalar data processing has variations depending on the number of finger touches and the variation is in the range of 1.3–1.7 ms. The touch controller with only the scalar processor needed 30.4 ms for one touch frame (1,200 nodes), with the data processing at a 100-MHz system clock frequency. The processing time of the touch controller with the accelerator was reduced to 5.6 ms at the same frequency. For the matrix filtering, it was reduced to 3.1 ms by the accelerator that is very fast and reduces repeatable memory readings and writings. Processing time in the accelerator was 0.05 ms when it was simulated by the Xilinx ISE Simulator (ISim, timing simulator) and it was hidden under the data transactions between the RISC core and the accelerator when we measured it. Processing time and the meaning of each functional block that is coded using C language in the RISC processor are listed on the Table 2.

Figure 6 shows the processing times of each algorithm in case of (a) the scalar processor only and (b) the scalar processor with the touch accelerator at a 100-MHz system frequency for the 10″ touch panel. With the touch accelerator,

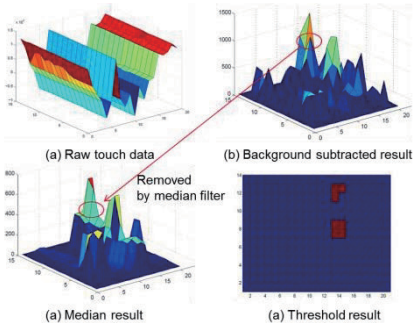

Fig. 5 Evaluation results of (a) raw touch data, (b) background subtraction, (c) median filter, and (d) threshold filter using the touch accelerator.

Table 1 Average processing times.

Panel Size	Configuration	Matrix filtering	Frame processing
5" (20 x 15)	Scalar only	7.3 ms	8.9 ms
	Scalar + accelerator	1.1 ms	2.7 ms
	Speed Up	***6.6×***	***3.3×***
10" (40 x 30)	Scalar only	27.9 ms	30.4 ms
	Scalar + accelerator	3.1 ms	5.6 ms
	Speed Up	***8.7×***	***5.4×***

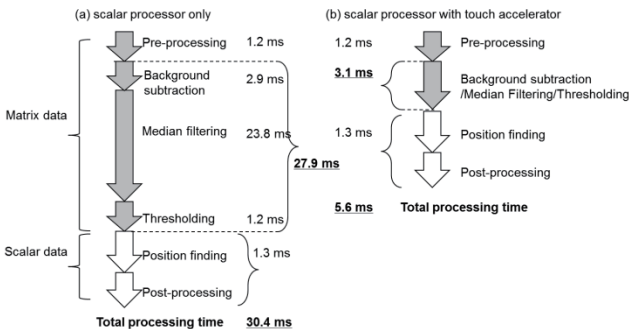

Fig. 6 Evaluation results of (a) the scalar processor only, and (b) the scalar processor with the touch accelerator.

the speed up was 8.7 times that of the scalar processor only, for matrix data filtering. For all of the touch processing, the touch controller completes each frame within in 5.6 ms. The touch controller achieved a 178-Hz reporting rate using the proposed touch accelerator only considering digital processing time. Total processing time was reduced from 30.4 ms to 5.6 ms, and its speedup was 5.4×.

Adding the touch accelerator increased 1 mW of dynamic power consumption at an FPGA, and power consumption was simulated by Xilinx XPower analyzer. For the estimation, the architecture of our touch accelerator was

Table 2 Processing time for the matrix filtering.

Meaning of functional block	Processing time of each function
loop controls for touch nodes	0.98 ms
Loading background/touch data & merging	1.14 ms
Writing data to / Reading results from the accelerator (included processing time in the accelerator)	0.48 ms
Storing results to the data memory	0.51 ms
Processing time for matrix filtering	***3.11 ms***

Table 3 Synthesis results of the proposed touch accelerator.

Clock constraint	Area	Power Consumption
2 ns	10,651 μm^2	391.9 μW
10 ns	10,651 μm^2	249.2 μW

also synthesized with the SAED 28-nm process, and the results are shown in Table 3. Total area was synthesized at 10,651 μm^2 and additional power consumption was only 249.2 μW at 10 ns clock constraint.

4. Conclusion

In this letter, we presented a real-time touch controller for a large-sized touch screen with a specialized touch processing accelerator and peripherals. For high-speed processing, a parallel processing algorithm was implemented, and its additional power consumption was only 250 μW at a 100-MHz clock frequency. Assuming a 10-inch touch panel that has 1,200 touch nodes, the touch controller can archive a 178-Hz report rate using the touch accelerator.

Acknowledgments

This research was financially supported by LG Display.

References

[1] S.S. Hwang, S.H. Bae, S.-H. Cho, H.-K. Kang, D.S. Lee, J.K. Shin, and I.J. Jung, "On-cell projected capacitive type touch sensor for NBPC," SID 2010 Digest, vol.45, no.3, pp.677–679, 2010.
[2] S.H. Bae, J. Park, C.S. Kim, S.W. Lee, W. Shin, and Y.-S. Lee, "Evaluation of large-sized LCD touch panel using differential sensing circuit and algorithm," IEICE Trans. Inf. & Syst., vol.E97-D, no.5, pp.1363–1366, May 2014.
[3] C.-Y. Lee, P.-W. Hsieh, and J.-M. Tsai, "High-speed median filter designs using shiftable content-addressable memory," IEEE Trans. Circuits Syst. Video Technol., vol.4, no.6, pp.544–549, 1994.
[4] R.-D. Chen, P.-Y. Chen, and C.-H. Yeh, "Design of an area-efficient one-dimensional median filter," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol.60, no.10, pp.662–666, 2013.